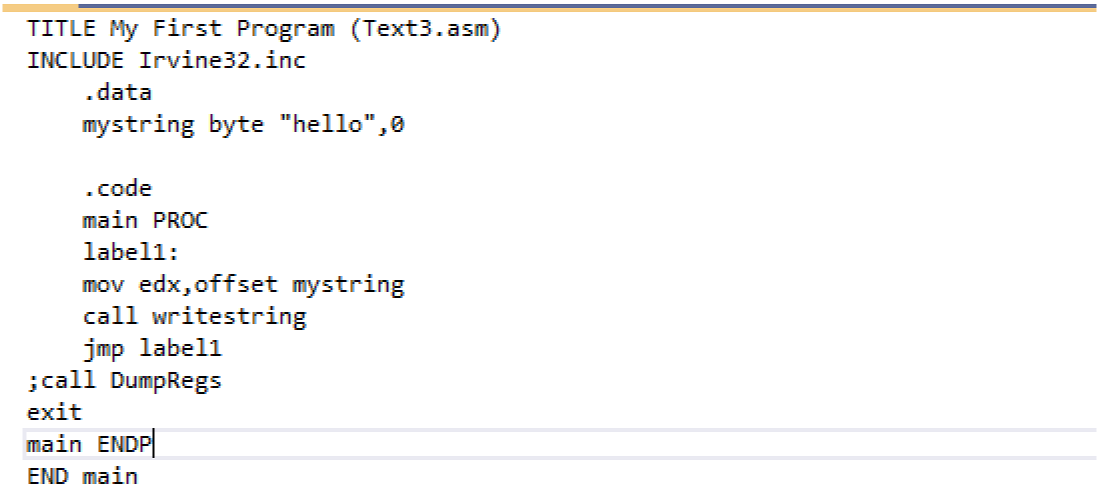
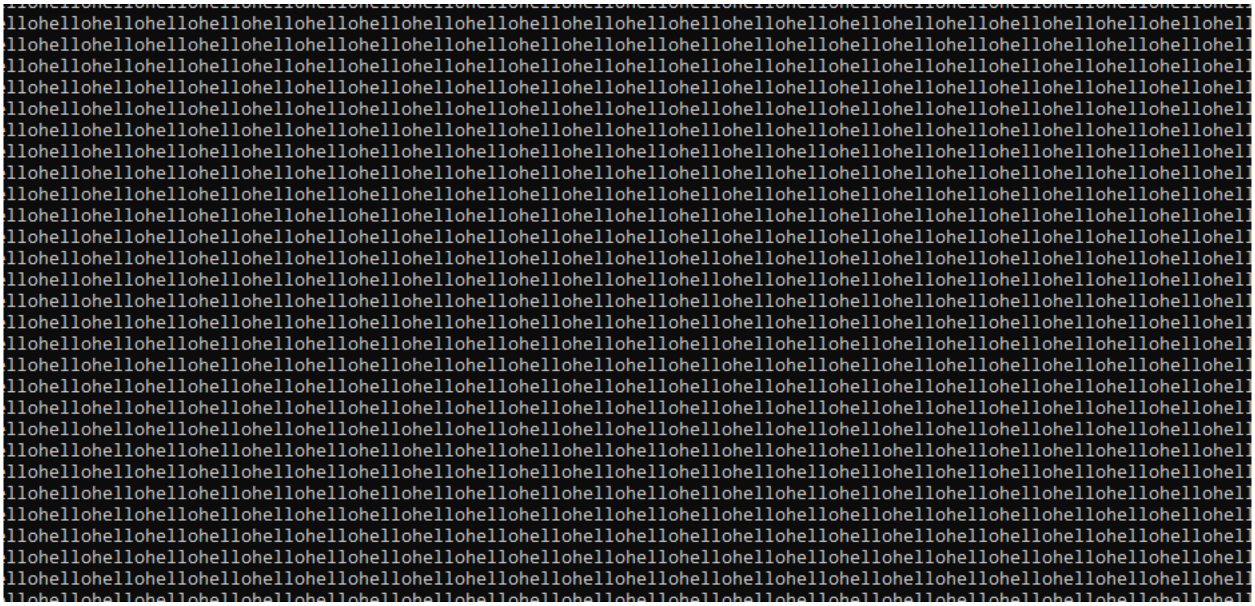
Examples

Example#1



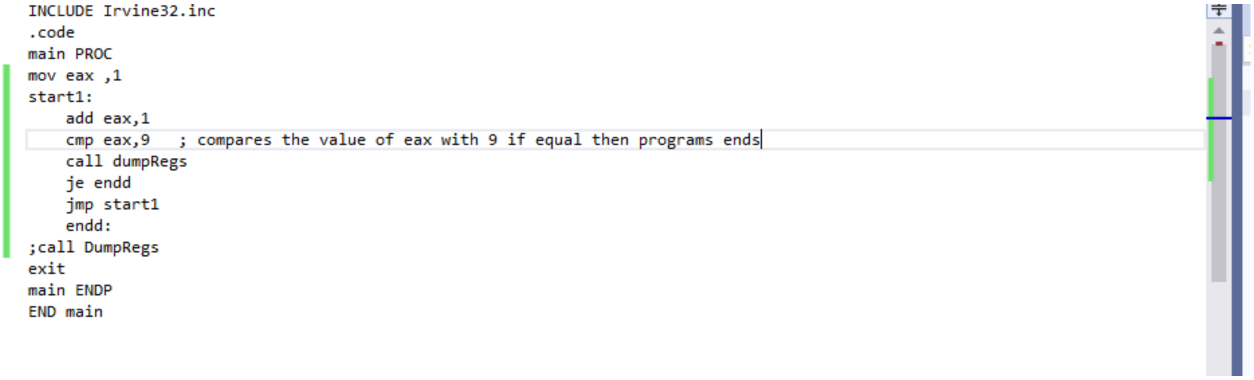
```
TITLE My First Program (Text3.asm)
INCLUDE Irvine32.inc
    .data
    mystring byte "hello",0

    .code
    main PROC
    label1:
    mov edx,offset mystring
    call writestring
    jmp label1
;call DumpRegs
exit
main ENDP
END main
```

OutPut



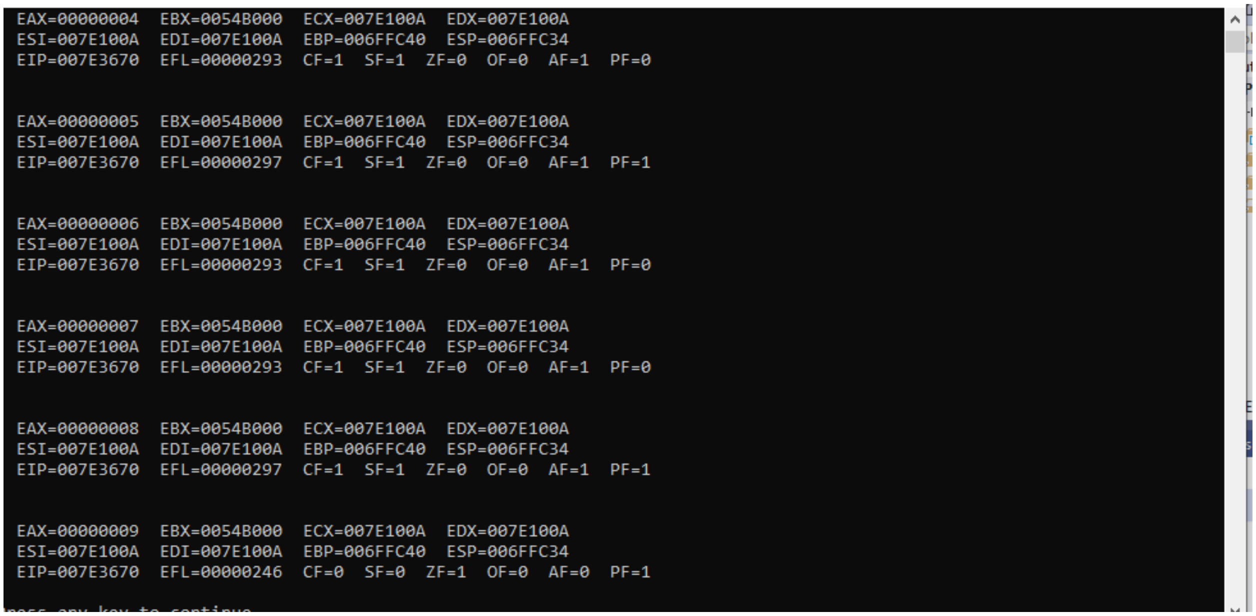
```
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
```

Example#2



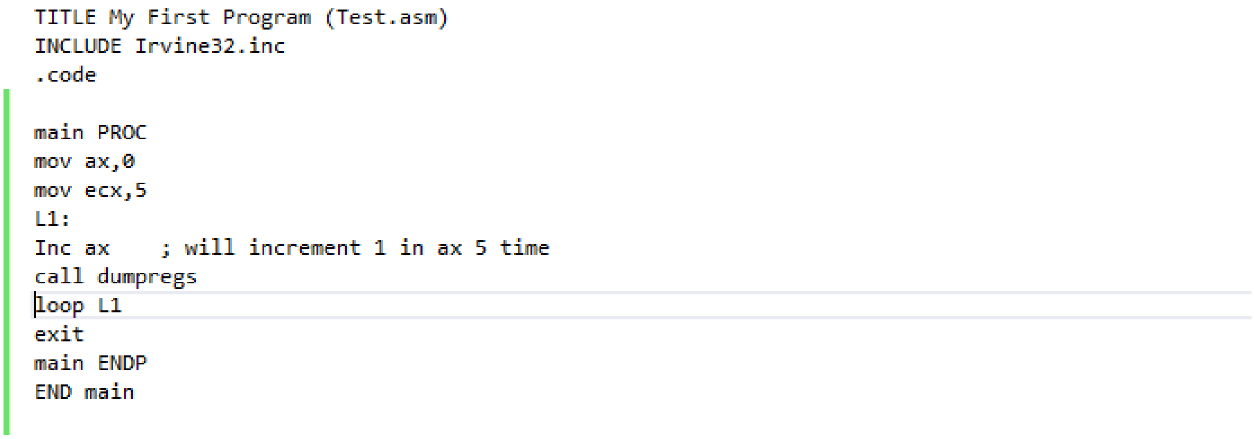
```
INCLUDE Irvine32.inc
.code
main PROC
mov eax ,1
start1:
    add eax,1
    cmp eax,9   ; compares the value of eax with 9 if equal then programs ends
    call dumpRegs
    je endd
    jmp start1
    endd:
;call DumpRegs
exit
main ENDP
END main
```

Output



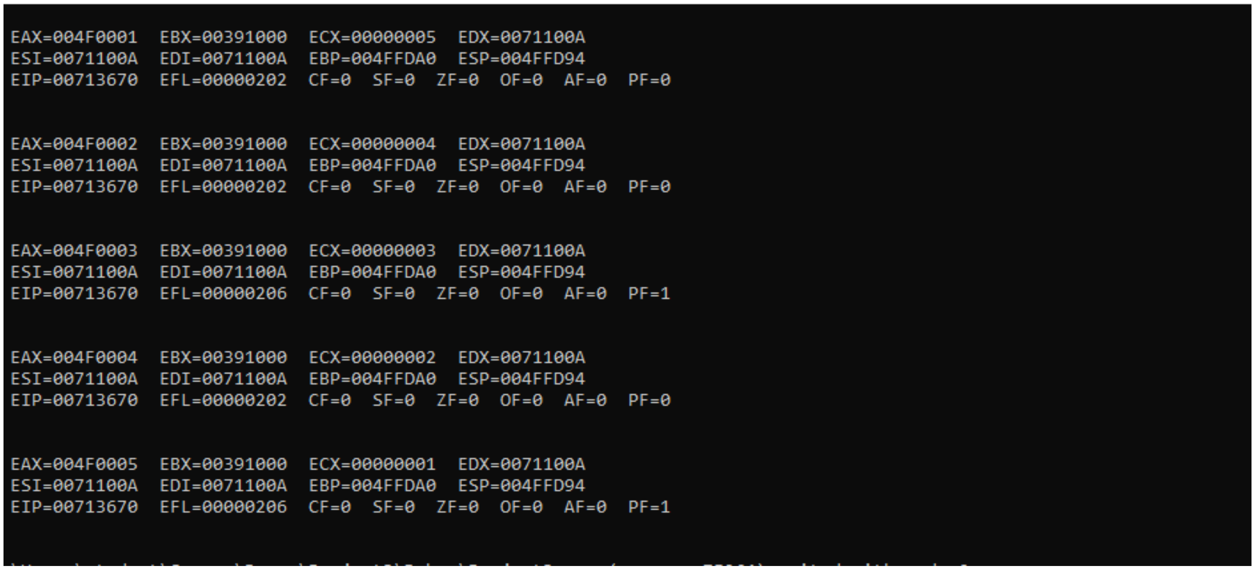
```
EAX=00000004  EBX=0054B000  ECX=007E100A  EDX=007E100A
ESI=007E100A  EDI=007E100A  EBP=006FFC40  ESP=006FFC34
EIP=007E3670  EFL=00000293  CF=1  SF=1  ZF=0  OF=0  AF=1  PF=0


EAX=00000005  EBX=0054B000  ECX=007E100A  EDX=007E100A
ESI=007E100A  EDI=007E100A  EBP=006FFC40  ESP=006FFC34
EIP=007E3670  EFL=00000297  CF=1  SF=1  ZF=0  OF=0  AF=1  PF=1


EAX=00000006  EBX=0054B000  ECX=007E100A  EDX=007E100A
ESI=007E100A  EDI=007E100A  EBP=006FFC40  ESP=006FFC34
EIP=007E3670  EFL=00000293  CF=1  SF=1  ZF=0  OF=0  AF=1  PF=0


EAX=00000007  EBX=0054B000  ECX=007E100A  EDX=007E100A
ESI=007E100A  EDI=007E100A  EBP=006FFC40  ESP=006FFC34
EIP=007E3670  EFL=00000293  CF=1  SF=1  ZF=0  OF=0  AF=1  PF=0


EAX=00000008  EBX=0054B000  ECX=007E100A  EDX=007E100A
ESI=007E100A  EDI=007E100A  EBP=006FFC40  ESP=006FFC34
EIP=007E3670  EFL=00000297  CF=1  SF=1  ZF=0  OF=0  AF=1  PF=1


EAX=00000009  EBX=0054B000  ECX=007E100A  EDX=007E100A
ESI=007E100A  EDI=007E100A  EBP=006FFC40  ESP=006FFC34
EIP=007E3670  EFL=00000246  CF=0  SF=0  ZF=1  OF=0  AF=0  PF=1

press any key to continue
```

Example #3



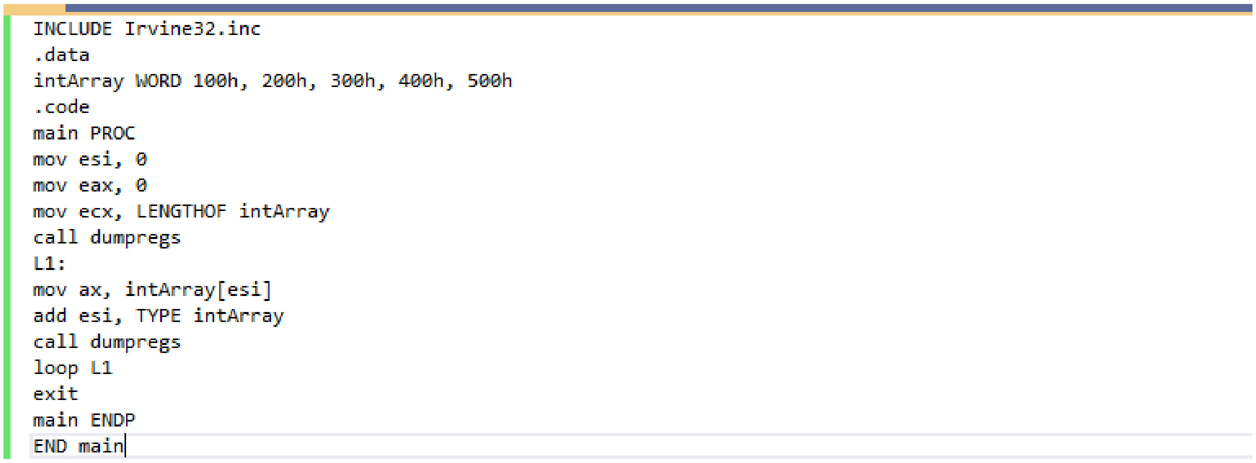
```
TITLE My First Program (Test.asm)
INCLUDE Irvine32.inc
.code

main PROC
mov ax,0
mov ecx,5
L1:
Inc ax     ; will increment 1 in ax 5 time
call dumpregs
loop L1
exit
main ENDP
END main
```

Output



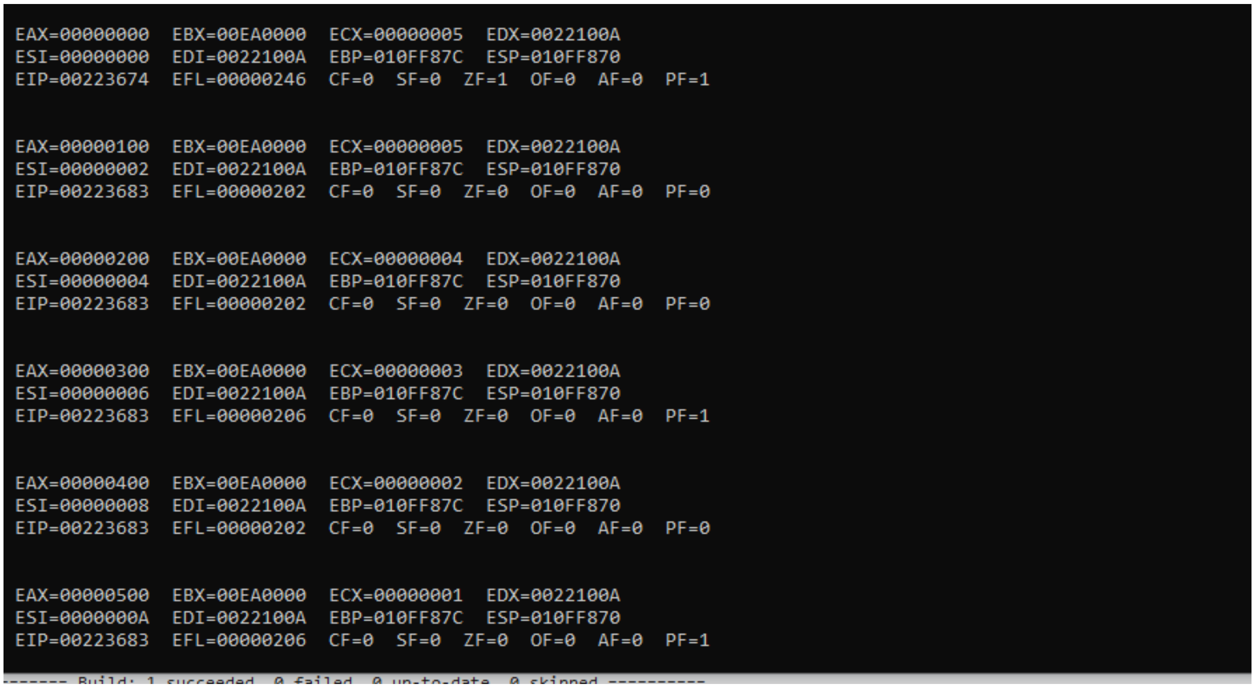
```
EAX=004F0001  EBX=00391000  ECX=00000005  EDX=0071100A
ESI=0071100A  EDI=0071100A  EBP=004FFDA0  ESP=004FFD94
EIP=00713670  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0


EAX=004F0002  EBX=00391000  ECX=00000004  EDX=0071100A
ESI=0071100A  EDI=0071100A  EBP=004FFDA0  ESP=004FFD94
EIP=00713670  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0


EAX=004F0003  EBX=00391000  ECX=00000003  EDX=0071100A
ESI=0071100A  EDI=0071100A  EBP=004FFDA0  ESP=004FFD94
EIP=00713670  EFL=00000206  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=1


EAX=004F0004  EBX=00391000  ECX=00000002  EDX=0071100A
ESI=0071100A  EDI=0071100A  EBP=004FFDA0  ESP=004FFD94
EIP=00713670  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0


EAX=004F0005  EBX=00391000  ECX=00000001  EDX=0071100A
ESI=0071100A  EDI=0071100A  EBP=004FFDA0  ESP=004FFD94
EIP=00713670  EFL=00000206  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=1
```

Example#4



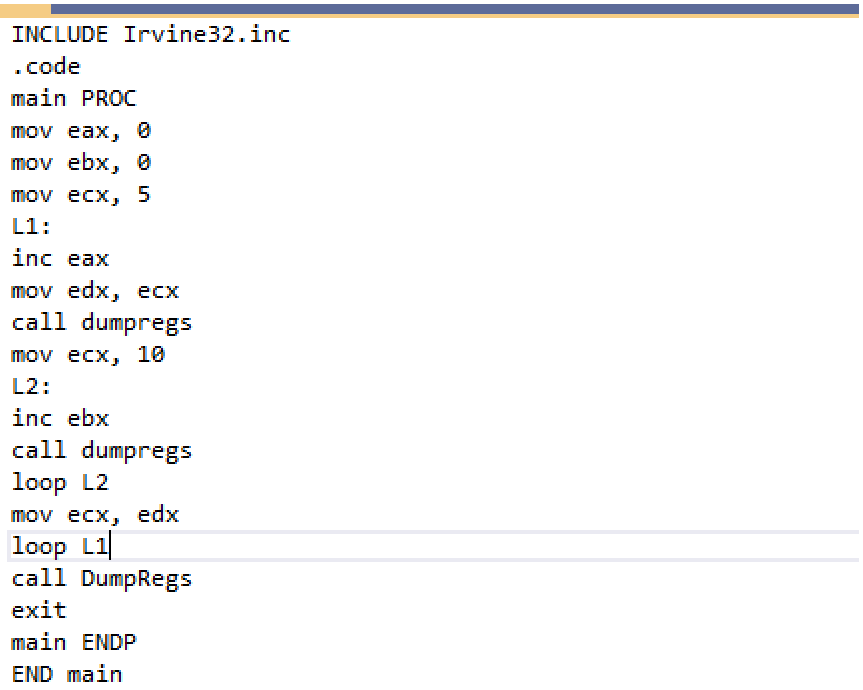
```
INCLUDE Irvine32.inc
.data
intArray WORD 100h, 200h, 300h, 400h, 500h
.code
main PROC
mov esi, 0
mov eax, 0
mov ecx, LENGTHOF intArray
call dumpregs
L1:
mov ax, intArray[esi]
add esi, TYPE intArray
call dumpregs
loop L1
exit
main ENDP
END main
```

Output



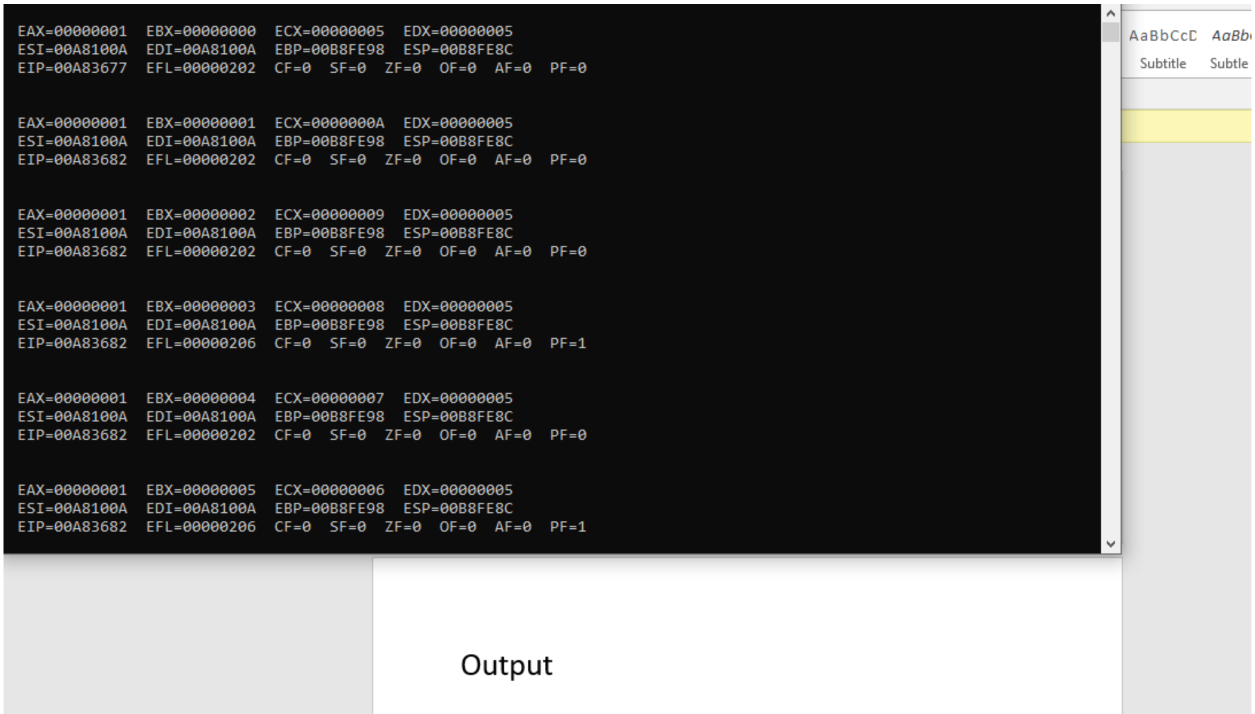
```
EAX=00000000  EBX=00EA0000  ECX=00000005  EDX=0022100A
ESI=00000000  EDI=0022100A  EBP=010FF87C  ESP=010FF870
EIP=00223674  EFL=00000246  CF=0  SF=0  ZF=1  OF=0  AF=0  PF=1


EAX=00000100  EBX=00EA0000  ECX=00000005  EDX=0022100A
ESI=00000002  EDI=0022100A  EBP=010FF87C  ESP=010FF870
EIP=00223683  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0


EAX=00000200  EBX=00EA0000  ECX=00000004  EDX=0022100A
ESI=00000004  EDI=0022100A  EBP=010FF87C  ESP=010FF870
EIP=00223683  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0


EAX=00000300  EBX=00EA0000  ECX=00000003  EDX=0022100A
ESI=00000006  EDI=0022100A  EBP=010FF87C  ESP=010FF870
EIP=00223683  EFL=00000206  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=1


EAX=00000400  EBX=00EA0000  ECX=00000002  EDX=0022100A
ESI=00000008  EDI=0022100A  EBP=010FF87C  ESP=010FF870
EIP=00223683  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0


EAX=00000500  EBX=00EA0000  ECX=00000001  EDX=0022100A
ESI=0000000A  EDI=0022100A  EBP=010FF87C  ESP=010FF870
EIP=00223683  EFL=00000206  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=1

------- Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ----------
```

Example#5



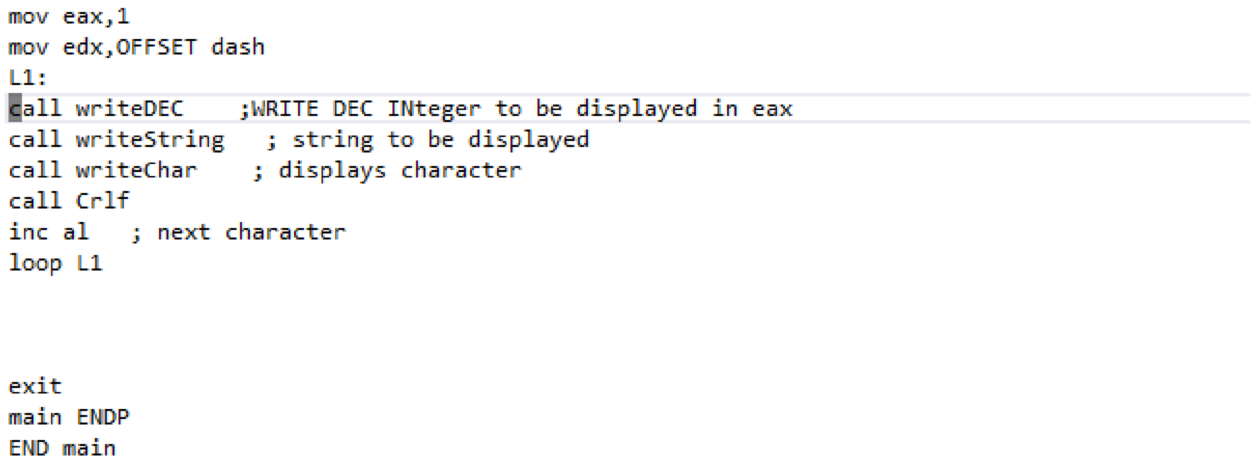
```
INCLUDE Irvine32.inc
.code
main PROC
mov eax, 0
mov ebx, 0
mov ecx, 5
L1:
inc eax
mov edx, ecx
call dumpregs
mov ecx, 10
L2:
inc ebx
call dumpregs
loop L2
mov ecx, edx
loop L1
call DumpRegs
exit
main ENDP
END main
```

Output



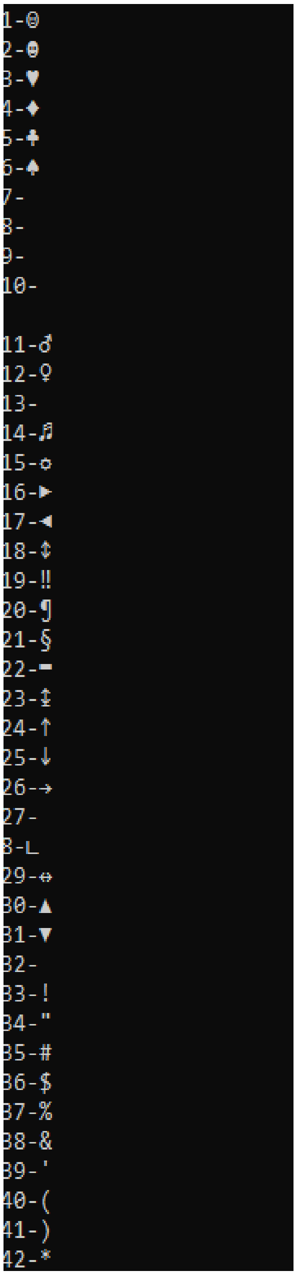
```
EAX=00000001  EBX=00000000  ECX=00000005  EDX=00000005
ESI=00A8100A  EDI=00A8100A  EBP=00B8FE98  ESP=00B8FE8C
EIP=00A83677  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0


EAX=00000001  EBX=00000001  ECX=0000000A  EDX=00000005
ESI=00A8100A  EDI=00A8100A  EBP=00B8FE98  ESP=00B8FE8C
EIP=00A83682  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0


EAX=00000001  EBX=00000002  ECX=00000009  EDX=00000005
ESI=00A8100A  EDI=00A8100A  EBP=00B8FE98  ESP=00B8FE8C
EIP=00A83682  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0


EAX=00000001  EBX=00000003  ECX=00000008  EDX=00000005
ESI=00A8100A  EDI=00A8100A  EBP=00B8FE98  ESP=00B8FE8C
EIP=00A83682  EFL=00000206  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=1


EAX=00000001  EBX=00000004  ECX=00000007  EDX=00000005
ESI=00A8100A  EDI=00A8100A  EBP=00B8FE98  ESP=00B8FE8C
EIP=00A83682  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0


EAX=00000001  EBX=00000005  ECX=00000006  EDX=00000005
ESI=00A8100A  EDI=00A8100A  EBP=00B8FE98  ESP=00B8FE8C
EIP=00A83682  EFL=00000206  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=1
```

Example#6

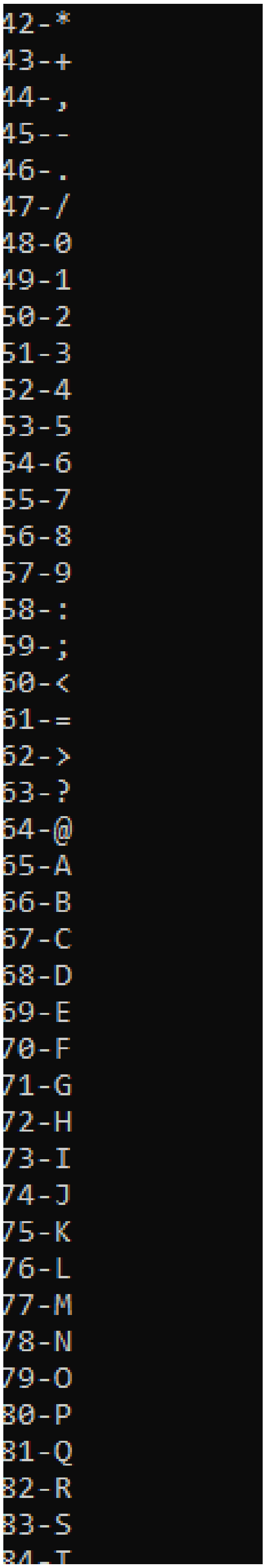


```
mov eax,1
mov edx,OFFSET dash
L1:
call writeDEC    ;WRITE DEC INteger to be displayed in eax
call writeString   ; string to be displayed
call writeChar    ; displays character
call Crlf
inc al   ; next character
loop L1



exit
main ENDP
END main
```

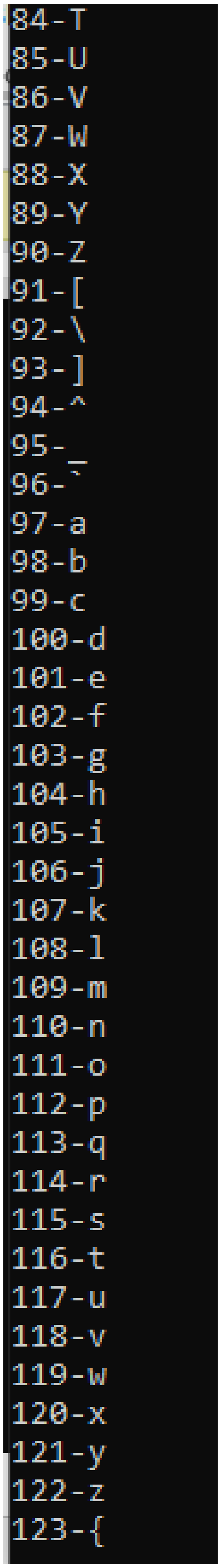
Output



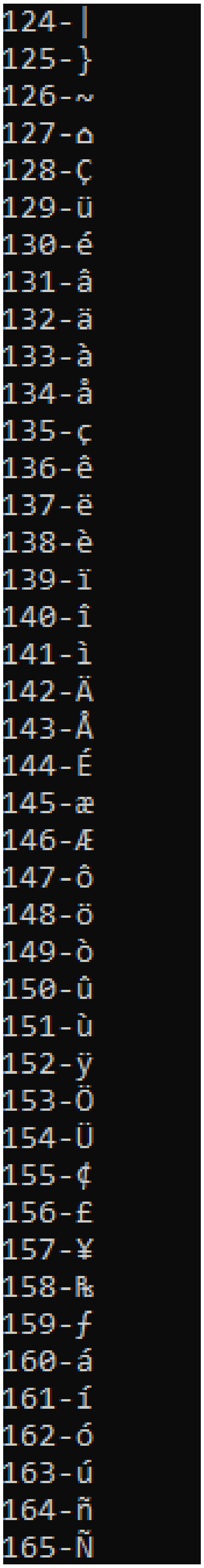
```
1-☺
2-☻
3-♥
4-♦
5-♣
6-♠
7-
8-
9-
10-

11-♂
12-♀
13-
14-♫
15-☼
16-►
17-◄
18-↕
19-‼
20-¶
21-§
22-▬
23-↨
24-↑
25-↓
26-→
27-
8-∟
29-↔
30-▲
31-▼
32-
33-!
34-"
35-#
36-$
37-%
38-&
39-'
40-(
41-)
42-*
```



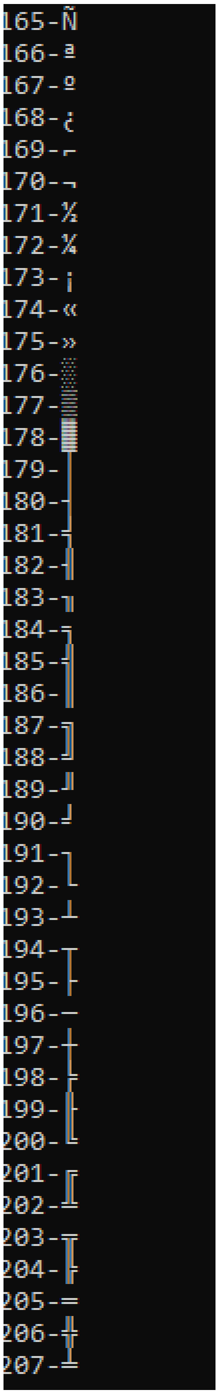
```
42-*
43-+
44-,
45--
46-.
47-/
48-0
49-1
50-2
51-3
52-4
53-5
54-6
55-7
56-8
57-9
58-:
59-;
60-<
61-=
62->
63-?
64-@
65-A
66-B
67-C
68-D
69-E
70-F
71-G
72-H
73-I
74-J
75-K
76-L
77-M
78-N
79-O
80-P
81-Q
82-R
83-S
```



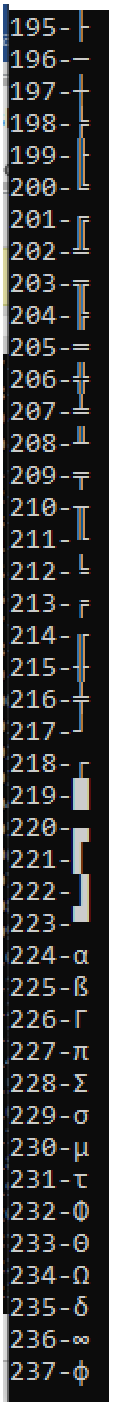
```
84-T
85-U
86-V
87-W
88-X
89-Y
90-Z
91-[
92-\
93-]
94-^
95-_
96-`
97-a
98-b
99-c
100-d
101-e
102-f
103-g
104-h
105-i
106-j
107-k
108-l
109-m
110-n
111-o
112-p
113-q
114-r
115-s
116-t
117-u
118-v
119-w
120-x
121-y
122-z
123-{
```



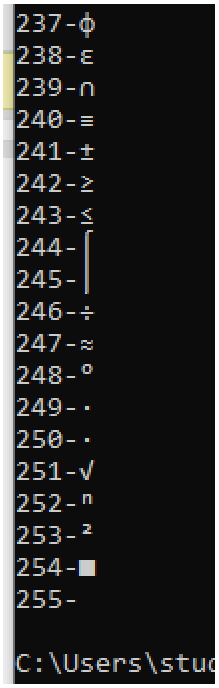
```
124-|
125-}
126-~
127-⌂
128-Ç
129-ü
130-é
131-â
132-ä
133-à
134-å
135-ç
136-ê
137-ë
138-è
139-ï
140-î
141-ì
142-Ä
143-Å
144-É
145-æ
146-Æ
147-ô
148-ö
149-ò
150-û
151-ù
152-ÿ
153-Ö
154-Ü
155-¢
156-£
157-¥
158-₧
159-ƒ
160-á
161-í
162-ó
163-ú
164-ñ
165-Ñ
```



```
165-Ñ
166-ª
167-º
168-¿
169-⌐
170-¬
171-½
172-¼
173-¡
174-«
175-»
176-░
177-▒
178-▓
179-│
180-┤
181-╡
182-╢
183-╖
184-╕
185-╣
186-║
187-╗
188-╝
189-╜
190-╛
191-┐
192-└
193-┴
194-┬
195-├
196-─
197-┼
198-╞
199-╟
200-╚
201-╔
202-╩
203-╦
204-╠
205-═
206-╬
207-╧
```

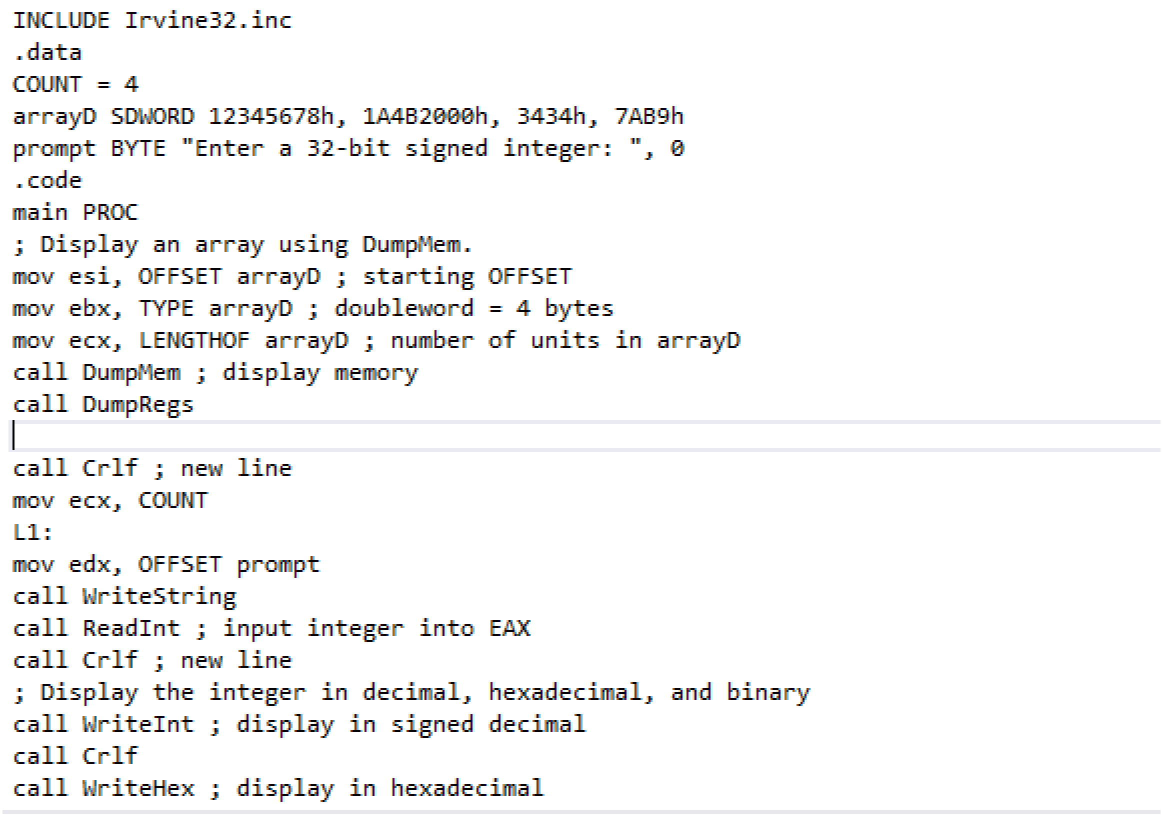


```
195-├
196-─
197-┼
198-╞
199-╟
200-╚
201-╔
202-╩
203-╦
204-╠
205-═
206-╬
207-╧
208-╨
209-╤
210-╥
211-╙
212-╘
213-╒
214-╓
215-╫
216-╪
217-┘
218-┌
219-█
220-▄
221-▌
222-▐
223-▀
224-α
225-ß
226-Γ
227-π
228-Σ
229-σ
230-µ
231-τ
232-Φ
233-Θ
234-Ω
235-δ
236-∞
237-φ
```

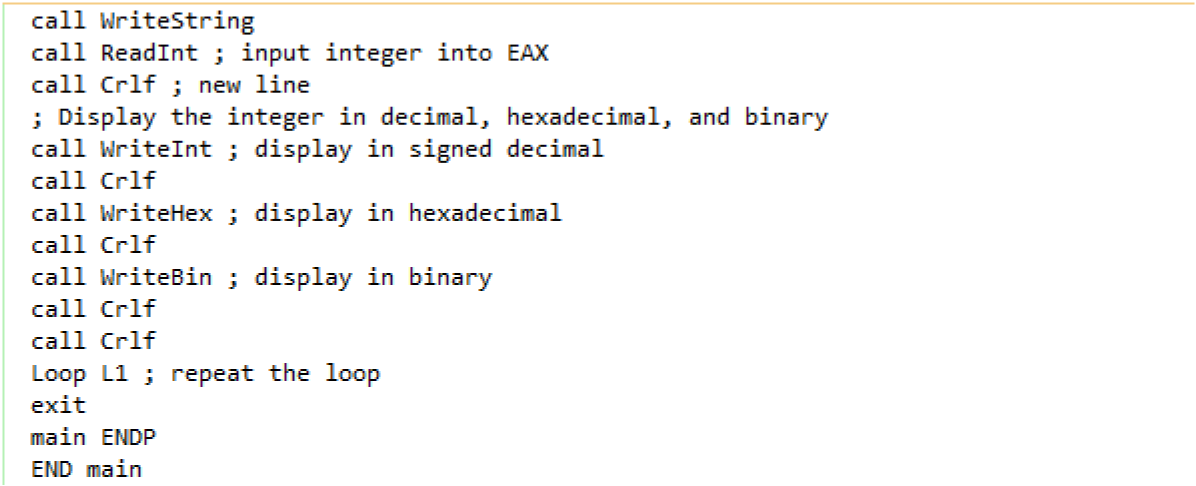


```
237-φ
238-ε
239-∩
240-≡
241-±
242-≥
243-≤
244-⌠
245-⌡
246-÷
247-≈
248-°
249-∙
250-·
251-√
252-ⁿ
253-²
254-■
255-

C:\Users\stuc
```

Example#7

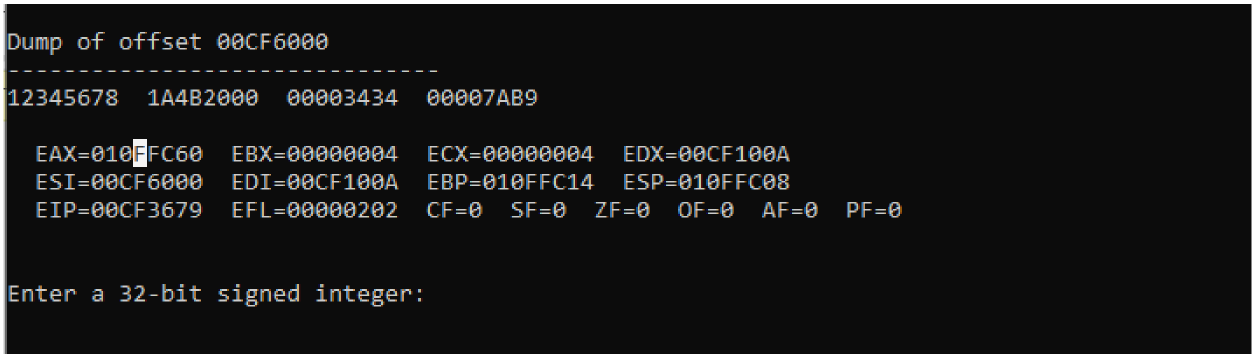


```
INCLUDE Irvine32.inc
.data
COUNT = 4
arrayD SDWORD 12345678h, 1A4B2000h, 3434h, 7AB9h
prompt BYTE "Enter a 32-bit signed integer: ", 0
.code
main PROC
; Display an array using DumpMem.
mov esi, OFFSET arrayD ; starting OFFSET
mov ebx, TYPE arrayD ; doubleword = 4 bytes
mov ecx, LENGTHOF arrayD ; number of units in arrayD
call DumpMem ; display memory
call DumpRegs

call Crlf ; new line
mov ecx, COUNT
L1:
mov edx, OFFSET prompt
call WriteString
call ReadInt ; input integer into EAX
call Crlf ; new line
; Display the integer in decimal, hexadecimal, and binary
call WriteInt ; display in signed decimal
call Crlf
call WriteHex ; display in hexadecimal
```



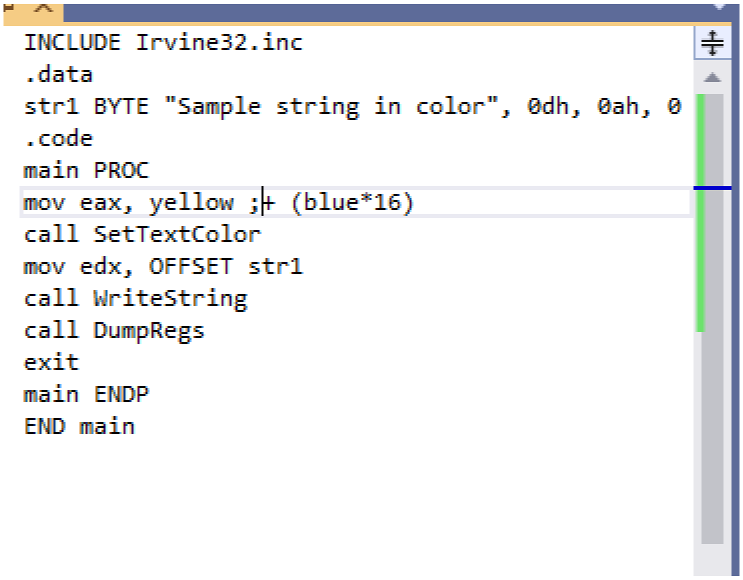
```
call WriteString
call ReadInt ; input integer into EAX
call Crlf ; new line
; Display the integer in decimal, hexadecimal, and binary
call WriteInt ; display in signed decimal
call Crlf
call WriteHex ; display in hexadecimal
call Crlf
call WriteBin ; display in binary
call Crlf
call Crlf
Loop L1 ; repeat the loop
exit
main ENDP
END main
```

Output



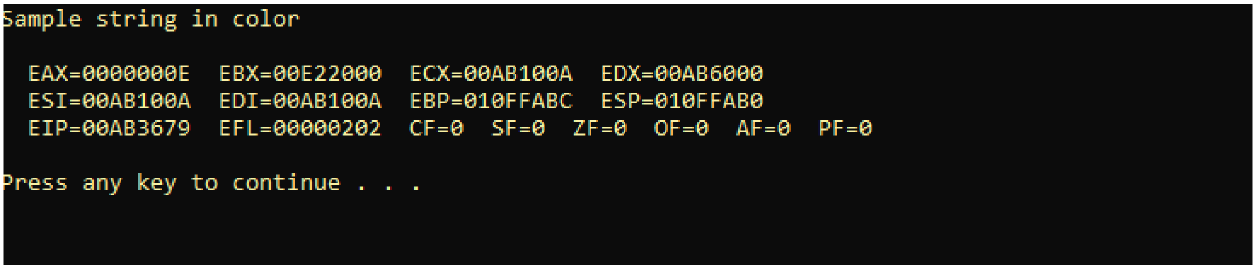
```
Dump of offset 00CF6000
-------------------------------
12345678  1A4B2000  00003434  00007AB9

  EAX=010FFC60  EBX=00000004  ECX=00000004  EDX=00CF100A
  ESI=00CF6000  EDI=00CF100A  EBP=010FFC14  ESP=010FFC08
  EIP=00CF3679  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0


Enter a 32-bit signed integer:
```

Example#8

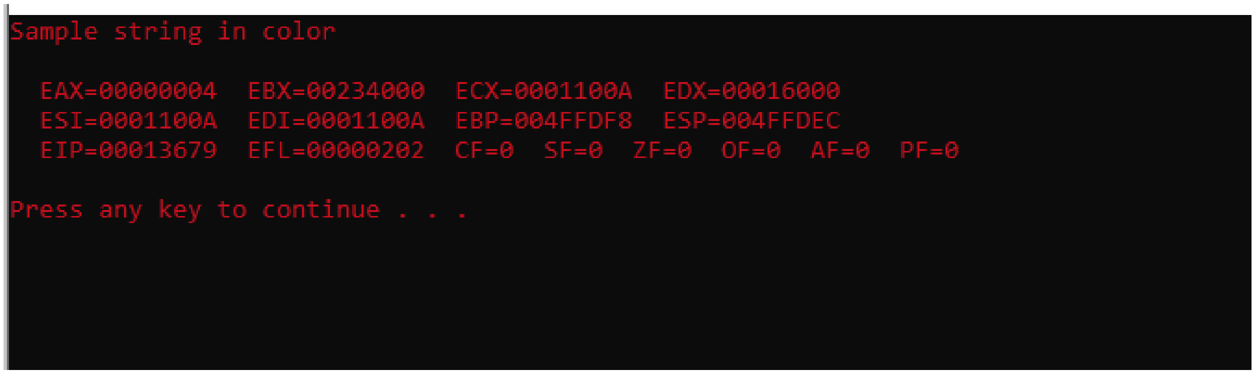


```
INCLUDE Irvine32.inc
.data
str1 BYTE "Sample string in color", 0dh, 0ah, 0
.code
main PROC
mov eax, yellow ;+ (blue*16)
call SetTextColor
mov edx, OFFSET str1
call WriteString
call DumpRegs
exit
main ENDP
END main
```

Output

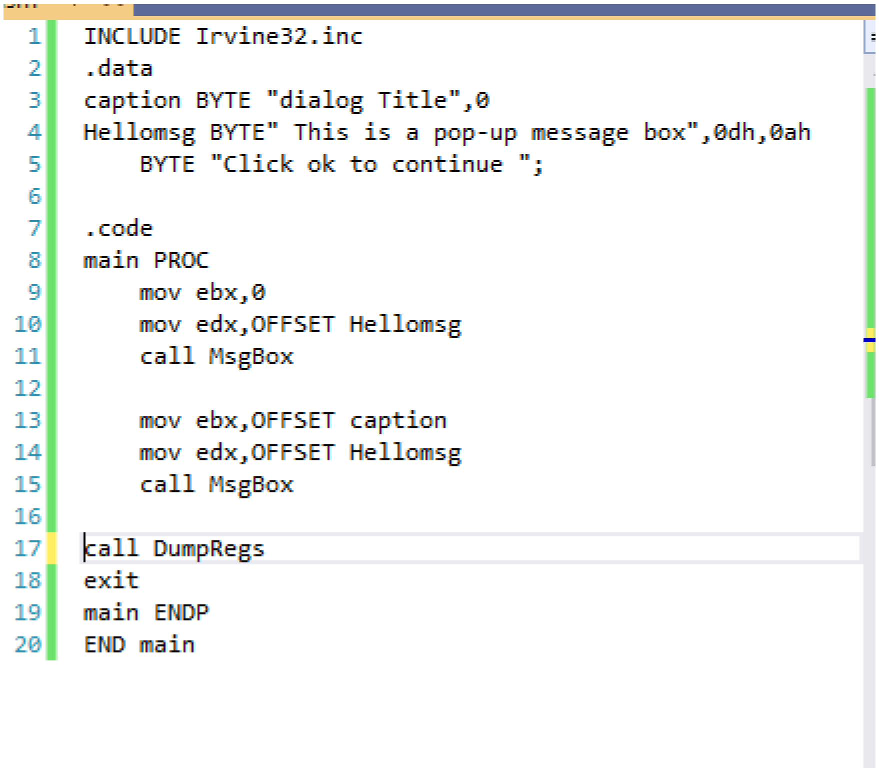


```
Sample string in color

  EAX=0000000E  EBX=00E22000  ECX=00AB100A  EDX=00AB6000
  ESI=00AB100A  EDI=00AB100A  EBP=010FFABC  ESP=010FFAB0
  EIP=00AB3679  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0

Press any key to continue . . .
```



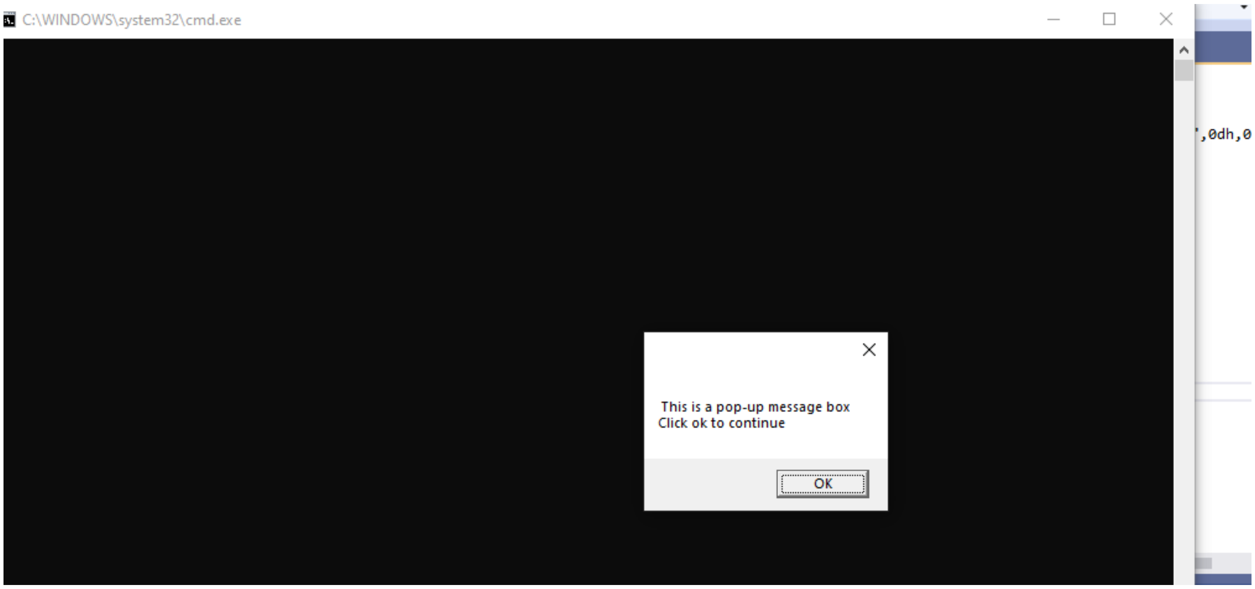
```
Sample string in color

  EAX=00000004  EBX=00234000  ECX=0001100A  EDX=00016000
  ESI=0001100A  EDI=0001100A  EBP=004FFDF8  ESP=004FFDEC
  EIP=00013679  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0

Press any key to continue . . .
```

Example#8

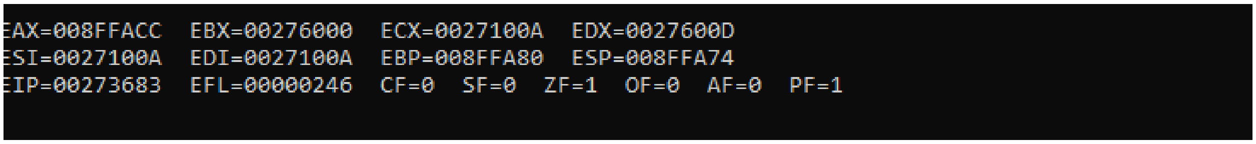


```
INCLUDE Irvine32.inc
.data
caption BYTE "dialog Title",0
Hellomsg BYTE" This is a pop-up message box",0dh,0ah
    BYTE "Click ok to continue ";

.code
main PROC
    mov ebx,0
    mov edx,OFFSET Hellomsg
    call MsgBox

    mov ebx,OFFSET caption
    mov edx,OFFSET Hellomsg
    call MsgBox

call DumpRegs
exit
main ENDP
END main
```

Output



```
This is a pop-up message box
Click ok to continue
```

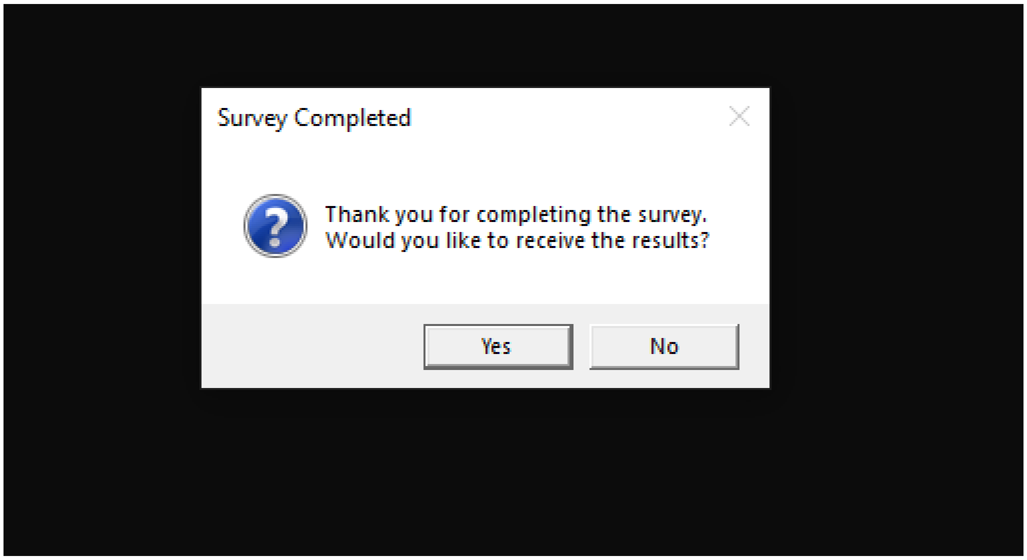


```
EAX=008FFACC  EBX=00276000  ECX=0027100A  EDX=0027600D
ESI=0027100A  EDI=0027100A  EBP=008FFA80  ESP=008FFA74
EIP=00273683  EFL=00000246  CF=0  SF=0  ZF=1  OF=0  AF=0  PF=1
```

EXAmple#9

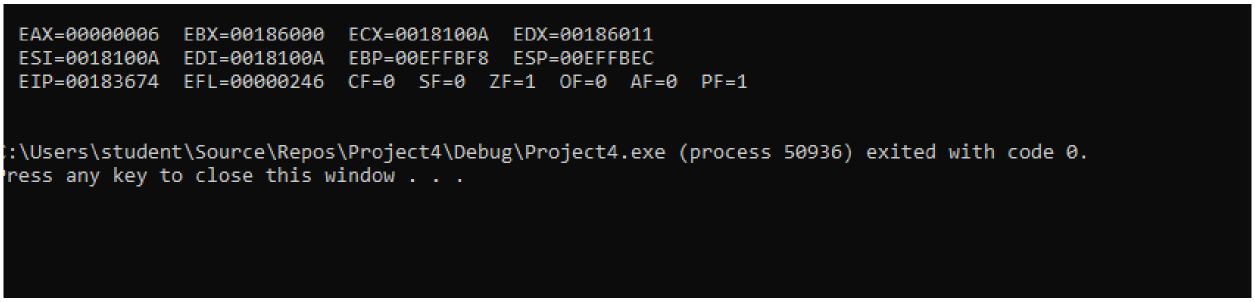
```
INCLUDE Irvine32.inc
;.data
.data
caption BYTE "Survey Completed",0
question BYTE "Thank you for completing the survey."
BYTE 0dh, 0ah
BYTE "Would you like to receive the results?", 0
.code
main PROC
mov ebx, OFFSET caption
mov edx, OFFSET question
call MsgBoxAsk



call DumpRegs
exit
main ENDP
END main
```

Output



Survey Completed

Thank you for completing the survey.
Would you like to receive the results?



```
 EAX=00000006  EBX=00186000  ECX=0018100A  EDX=00186011
 ESI=0018100A  EDI=0018100A  EBP=00EFFBF8  ESP=00EFFBEC
 EIP=00183674  EFL=00000246  CF=0  SF=0  ZF=1  OF=0  AF=0  PF=1


:\Users\student\Source\Repos\Project4\Debug\Project4.exe (process 50936) exited with code 0.
ress any key to close this window . . .
```